

FIELD

The present invention relates to multi-channel SONET/SDH desynchronizer for reading and transmitting asynchronous data received from several synchronous channels.

10

BACKGROUND

Data carried over a number of tributories such as DS1, DS3, E1, E3, etc., each of generally different frequencies, may be multiplexed together so that it can share a common transmitter, receiver and transmission media. In order to transmit such data together, it is mapped onto a frame at the place where it enters the network. All of the signals from the various tributories are synchronized to a single network clock. Excess bandwidth is filled with stuff bits which carry no information. A pointer adjustment can be used in place of or, in addition to, the stuff bits to indicate changes in the data rate of the synchronous channel and lack of synchronism of the nodes in the network. At the other end of the network a desynchronizer reads and transmits asynchronous data that had been carried over the synchronous channel. Variations in the instantaneous rate of the payload data caused by the interspersing of overhead data and the stuff bits can give rise to payload mapping jitter which can feed through into the desynchronizer output. The pointer adjustment due to lack of synchronism of the nodes in the network gives rise to another form of jitter which is more serious than mapping jitter. One pointer adjustment results in 8 cycles of

5 network clock jitter. The pointer adjustment occurrence is a stochastic process and, as a result, it is difficult to predict its behaviour. Desynchronization is used at the destination of the digital signals to also attenuate the mapping jitter. Mapping jitter is deterministic and easy to filter even with wide
10 bandwidth phase locked loop (PLL) circuits.

Desynchronizers typically include a first in first out buffer (FIFO), a phase detector to measure the FIFO buffer depth, a passive or active analog loop filter to filter the phase
15 detector output, and a voltage controlled oscillator (VCO) to generate an output clock to control data transmission from the FIFO buffer. These elements make up a phase locked loop.

There are some important differences in the way single
20 and multiple channel desynchronizers are designed. The term "multiple channel" means that more than one data signal is being dropped or extracted from the SONET/SDH signal. For example, in the case of the OC-1 signal one can drop 28 DS1 signals, which might need to be desynchronized. In the case of the OC-3 signal
25 one can drop 3 DS3 signals or 84 DS1 signals. In the case of the OC-12 one can drop up to 12 DS3 signals or 336 DS1 signals. With the advance of data communications and the requirement for higher bandwidths, there are more and more instances of multiple channels being dropped from the same SONET/SDH signal.

5 A multiple channel desynchronizer should not use a
voltage-controlled oscillator (VCO) to generate the
desynchronized output clock. Most of the known desynchronizers
use voltage-controlled crystal oscillators (VCXOs) or VCOs in
cases where desynchronizers produce a control voltage that
10 controls the frequency of the VCO. Multiple channels, for
example 12 channels of DS3 can be dropped from OC12, can have
output frequencies quite close to each other and, as a
consequence, mutual coupling can cause excessive jitter. The
reason for the excessive jitter is caused by the large index of
15 modulation of the frequency modulation device. The index of
modulation of the frequency modulation device is inversely
proportional to the frequency of modulation. Thus, small
frequency offsets between desynchronized clocks, if there is even
a very small coupling between the desynchronization circuits,
20 would produce through PLL action very low modulation frequencies
on the control voltages which would modulate, excessively, the
VCO outputs.

 The multiple channel desynchronization circuit should
25 use a phase modulation method when generating desynchronized
clocks to avoid the excessive jitter caused by mutual coupling of
output clocks. The phase modulation device can be implemented as
an endless phase modulator, a numerically controlled oscillator
(NCO) or a single side-band modulator (SSB). Several patents
30 have already suggested use of the NCO devices for the
desynchronization. However, a large number of the NCO devices on

5 the same chip, especially if generating high frequencies, would
result in large power consumption. A better way, as far as power
consumption is concerned, where an NCO is used to synthesize a
low frequency, is to up-convert this low frequency using a mixer
and a high frequency local oscillator (LO). The output signal
10 from the mixer is filtered using the LC filter. In an ASIC
implementation of the desynchronizer use of the LC filter is not
an option. More suited for the ASIC implementation is a version
of up-conversion implementation, which uses the SSB modulator.
By the careful matching of phases and amplitudes in two branches
15 containing double balanced mixers, the SSB modulator suppresses
unwanted products of mixing, the LO leakage and the undesired
side-band. Normally, the SSB modulators can suppress the
unwanted products by more than 20 dBs, resulting in jitter of
less than 0.05 UIpp. The SSB modulation results in intrinsic
20 jitter well below the level allowed by Bellcore GR-253-CORE.
Some care has to be taken to avoid excessive mutual coupling of
output clocks because they still add to jitter, although there is
no excessive problem with low frequency offsets as in the case of
the frequency modulation device.

25

The preferred implementation would use an endless phase
modulator, because it is a digital approach and it is robust, due
to the fact that coupling between different output signals has
the least effect.

30

5 The second conclusion from investigating a multiple
channel desynchronizer is that a separate desynchronizer for each
channel will result in an extremely large circuit size. That is
specifically true if each of the necessary steps to be performed
are implemented without the re-use of circuitry. None of the
10 prior art discloses the re-use of the circuitry. Accordingly, it
is an object of the invention to merge the desynchronization of
all channels into a single channel by re-using the
desynchronization circuitry.

15 **SUMMARY OF THE INVENTION**

 According to the invention there is provided a single
desynchronizer for desynchronizing a plurality of data channels
of data signals by desynchronizing each channel in turn. The
desynchronizer has a plurality of first-in first-out buffer
20 (FIFO) blocks, one for each of said data channels. The FIFO
blocks each have respective FIFO read and write address outputs
and a gapped clock input operative in response to gapped clock
signals, to store input data extracted from a SONET/SDH frame.
An arithmetic unit has a phase word output and inputs coupled to
25 the read and write address outputs of each of the FIFO blocks,
and is operative to calculate an address difference of the read
and write addresses for each of the FIFO blocks and a phase
locked loop (PLL) phase increment value. The PLL value depends
on the address difference. A pointer adjustment phase difference
30 is added or subtracted from the PLL phase increment value to
produce a total phase increment from said phase word output for

5 each of said FIFO blocks. An endless phase modulator common
control block is coupled to an output of the arithmetic unit and
is operative to produce delay tap control signals in response to
corresponding total phase increment signals from the arithmetic
control unit. An endless phase modulator is coupled to an output
10 of the endless phase modulator common control block and to an
oscillator and is operative in response to the delay tap control
signals to generate clock signals frequency shifted from the
oscillator clock frequency and to apply these clock signals to
respective desynchronized clock inputs of said FIFO buffer
15 blocks. In this way desynchronized data signals are clocked out
from the FIFO buffer blocks.

The endless phase modulator may include a delay line
having delay elements with taps at junctions of the delay
20 elements and a plurality of modulator multiplexers, one for each
of the plurality of data channels, coupled to the taps and
operative to select a delay tap control signal in response to
receipt of a tap select signal.

25 Preferably, the delay elements are buffers.

An arithmetic unit, which is coupled to the FIFO,
performs all operations required for single or multi-channel
desynchronization. The endless phase modulator is coupled to the
30 arithmetic unit and to the FIFO and is operative, in response to

5 input from the arithmetic unit, to produce a single output
desynchronized clock or multiple output desynchronized clocks.

Several steps are necessary in the desynchronization process. Depending on which algorithm is being used, some steps
10 are optional. A desynchronizer is a form of phase locked loop (PLL). A FIFO is used to store data clocked into the FIFO with a gapped clock, which is a result of extracting a digital signal from a SONET/SDH frame. A narrow-band PLL is used to average this gapped clock with a long time-constant and clock the data
15 out of the FIFO with a non-jittery clock, whose frequency is equal to the average frequency of the gapped clock. Firstly, there is a need to form a difference between the FIFO write and read addresses. Forming a difference requires a subtract circuitry. The width of the subtract circuitry depends on the
20 FIFO size which, on the other hand, depends primarily on the data rate of the channel being desynchronized. This is equivalent to building a phase detector for the PLL operation. This subtract action forms the error signal for the PLL.

25 The next step is to filter the FIFO write and read address difference to average the in-flux of data into the FIFO and make it even with the out-flux of the data from the FIFO. The previous state of the art uses separate circuitry for filtering whereas the present desynchronization circuit re-uses
30 the same add-subtract circuit to detect an error signal for the PLL and also to perform the filtering. A filter with a perfect

5 integrator and a zero in a transfer function centers the FIFO
fill level for any steady state type of stimulus. This type of
loop filter has a superior performance as compared to the filter
without the perfect integrator. Most of the state of the art
desynchronizers use the latter kind of loop filter. Next, most
10 desynchronizers use separate circuitry to process the pointer
adjustment related bits in the FIFO, in many cases building a
separate, very often, adaptive filter for this purpose, whereas
the present circuit reuses the same add-subtract circuit to
achieve the same effect. In processing the pointer adjustment
15 related bits in the FIFO, the present desynchronizer goes a step
further than known desynchronizers because the present one uses a
digitally controlled endless phase modulator to perform a semi-
open loop modulation of the endless phase modulator. Most known
desynchronizers use bit leaking in which they hide the pointer
20 adjustment related bits that are in the FIFO from the PLL. Using
some algorithm for filtering, these bits are re-introduced to the
PLL. The embodiment of the multi-channel desynchronizer
disclosed can easily be changed to perform the same function, but
superior performance can be achieved if one uses the semi-open
25 loop modulation of the endless phase modulator. In this case,
the undesirable response of the PLL is eliminated when an error
signal is introduced, and the pointer adjustment related bits in
the FIFO are completely hidden from the PLL. The state of the
art desynchronizers that do not use a digitally generated clock
30 can not completely hide the pointer adjustment related bits in
the FIFO from the PLL; such desynchronizers do it only

5 temporarily for the incoming bits, but need to re-introduce the incoming bits to the PLL because they do not have enough precision to synthesize the desired clock. Such prior art desynchronizers must depend on the PLL to adjust for any inaccuracy in synthesis. The state of the art desynchronizers,
10 that use a digitally generated clock, can perform arbitrary phase modulation of the generated clock, but only U.S. Patent No. 5,497,405 issued to Elliot et al. uses this technique. Unfortunately, the method used in Elliot et al. creates unnecessary jitter because it uses only open loop. The present
15 method needs to use, for its reference, a clock identical to the add side system clock. This is not easy to achieve, as even then it produces unnecessary jitter for each stuffing bit. On the other hand, the present embodiment makes it is easy to implement, re-using the same circuitry, the semi-open loop for handling the
20 pointer adjustment related jitter and closed loop for handling bit mapping and network related jitter. Because digital modulation of an endless phase modulator is used, one knows exactly how many bits came to the FIFO due to the pointer adjustment and how many have leaked out by modulating the endless
25 phase modulator in an open loop fashion, so the bits in the FIFO resulting from the pointer adjustment can be hidden from the PLL at all times. The small inaccuracy of this process resulting from the small difference between the line clock used to generate the modulation that leaks the pointer adjustment related bits
30 from the FIFO, and the actual desynchronized clock frequency is taken care by the PLL. However, this inaccuracy has an extremely

5 small error so it does not produce any undesirable response of
the PLL.

10 The present invention includes a novel embodiment of a
desynchronizer for single or the multiple data channels dropped
from the SONET/SDH signal. It uses compact circuitry capable of
processing a number of channels that requires only addition or
subtraction circuits, multiplex circuits and RAM. This circuitry
is re-used to perform necessary operations for a single channel
desynchronizer and as well for a multiple channel desynchronizer.

15 The circuit consists of a FIFO block or several FIFO
blocks for individual data channels. All processing is done in
one arithmetic unit block and an output desynchronized clock or
clocks, in the case of multi-channel applications, are produced
in an endless phase modulator/modulators. Alternatively, one can
20 substitute for the endless phase modulators SSBs (single sideband
modulators) or, in case of low frequency data channels (DS1, E1),
with digitally controlled programmable modulo dividers.

25 The same arithmetic unit circuit can be used to execute
all operations necessary for a single or multi-channel
desynchronizer using a narrow-band second order type two PLL with
adjustable loop bandwidth to avoid FIFO spill while
simultaneously independently processing and filtering pointer
30 adjustment related bits in FIFO.

5 The same arithmetic unit circuit can be used to execute
all operations necessary for a single or multi-channel
desynchronizer using a narrow-band second order type dual PLL
with adjustable loop bandwidth to avoid FIFO spill while
simultaneously, independently processing adjustment related bits
10 in the FIFO using an open loop method of modulating the endless
phase modulator.

15 The proposed endless modulator block is a novel way to
generate multiple clocks at different frequencies using one delay
line and one calibration circuit. Novel use of a look-up list
(LUT), built using a RAM, that converts the instantaneous phase
value to the select signal for each of the multiplexer circuits
that generate different clocks, results in a loop bandwidth
independent of the number of taps whose delay is equivalent to
20 one clock pulse. Use of a dual port RAM and calibration circuit
reduce the size of the LUT. This method results in a PLL loop
bandwidth independent of process variation and temperature. Use
of the endless phase modulator significantly reduces power
consumption and gate count compared to a numerically controlled
25 oscillator (NCO) method.

BRIEF DESCRIPTION OF THE DRAWINGS

Further features and advantages will be apparent from
the following detailed description, given by way of example, of a
30 preferred embodiment taken in conjunction with the accompanying
drawings, wherein:

5

Fig. 1 is a multi-channel desynchronizer;

Fig. 2 is a schematic diagram of an arithmetic unit
block diagram;

10

Fig. 3 is a schematic diagram of a memory map of
arithmetic unit RAM blocks;

15

Figs. 4-11 are schematic diagrams of steps 1-8,
respectively, of the arithmetic unit algorithm;

Fig. 12 is an endless phase modulator common control
block;

20

Fig. 13 is an endless phase modulator delay line;

Fig. 14 is a delay line calibration circuit;

25

Fig. 15 is an alternative embodiment of the
desynchronizer using a single side-band modulator (SSB);

Fig. 16 is a waveform diagram of the calibration
circuit for both a low and a high tap number;

30

Fig. 17 is a graph of the selected tap number versus
address for the case the latched tap number is 15; and

Fig. 18 is a graph of the selected tap number versus address for the case where the latched tap number is 20.

DETAILED DESCRIPTION WITH REFERENCE TO THE DRAWINGS

Referring to Figure 1, the desynchronizer includes a FIFO block 12, which receives synchronized data from a set of SONET/SDH data lines 14 and gapped clock signals on gapped clock lines 16. An arithmetic unit block 18 is coupled to the FIFO block 12 by both a write address block 20 and a read address block 22. An endless phase modulator block 24 couples to the arithmetic unit block 18 and to a crystal oscillator 26. The crystal oscillator 26 oscillates at a nominal data rate of the digital signal which, for example, for desynchronizing DS3 is a rate of 44.736 MHz. The output of this crystal oscillator 26 is used in the endless phase modulator block 24 to generate the desynchronized clocks on desynchronized clock lines 28. A delay line with taps and a multiplex circuit that selects a specific tap generates different output clocks (see Fig. 13). Successive selecting of a higher and higher tap number of the delay line creates a lower frequency clock than the crystal oscillator frequency. Successive selecting of lower and lower tap numbers of the delay line creates a clock of higher frequency than the crystal oscillator frequency.

The arithmetic unit (AU) block 18 is common for all the channels and it uses a simple structure that performs a number of

5 different functions. The Block Diagram of the AU is shown in
Figure 2.

The arithmetic unit block **18** consists of one wide add-
subtract circuit **34**, a register **36** that latches the product of
adding or subtracting and preferably two RAM blocks **38** and **40**
10 that hold information used in calculating a phase increment for
each channel. The memory map of these two RAM blocks is shown in
Figure 3. It is possible to use one RAM block, but the speed of
operation is much faster with two RAM blocks. The first RAM
block **38** contains the effective FIFO address difference from
15 which the bits related to the pointer adjustment in the FIFO **12**
have been subtracted. This RAM block can also contain constant
values used in processing the number of bits in the FIFO **12** due
to the pointer adjustments. The second RAM block **40** contains the
values of the accumulated effective FIFO address difference, as
20 well as the current number of bits in FIFO **12** that are related to
the pointer adjustments. These pointer adjustment related bits
in FIFO **12** are slowly disposed of by adding an extra amount of
phase increment to calculated phase increments resulting from PLL
calculations. This extra phase increment, that leaks pointer
25 adjustment bits from the FIFO **12**, is independent of the
calculated phase increment for PLL operation and it is added to
the PLL phase increment after the PLL calculation. The pointer
adjustment bits in the FIFO **12** are handled in such a matter to
deny the PLL the knowledge of their existence.

5 The arithmetic unit block **18** contains three multiplex
circuits. Two multiplex circuits, MUX #1 **42**, and MUX #2 **44** are
used to provide the add-subtract circuit **34** with the desired
input, either the FIFO address for each of the channels or the
RAM output. The third multiplex circuit, MUX #3 **46**, scales the
10 output of the operations, therefore it functions as a fast
multiply/divide by 2^m circuit, where m is an integer. An
adjustment of the PLL's Open Loop Gain and the Loop Filter
transfer function zero positioning can be set using multiplex
circuit MUX #3 **46**.

15 The control circuit **48** conducts the operation of the
arithmetic unit block **18**. It is designed as a number of counters
(not shown) that step the desynchronizer through the desired
phases. A simple change of the counters can be used to change
the algorithm used in the desynchronization.

20 The arithmetic unit block **18** runs synchronously with a
51.84MHz clock recovered from the lines **16**. The SONET/SDH frame
row rate is 72 kHz. The FIFO write **20** and read **22** addresses of
one of the channels are sampled at the 72 kHz rate and they are
processed at this rate. The sampling of the FIFO addresses can
25 be done at slower rate (example 8 kHz), but sampling of the FIFO
addresses at the highest possible rate at which reading of FIFO
write address **20** results in small mapping jitter, will minimize
the aliasing of jitter introduced over the optical line from the
higher frequencies into the loop bandwidth.

5 The operations that are required to process one channel are relatively simple and can be sequenced one after other using the same basic circuitry. Operations are:

- Subtract read FIFO address from the write FIFO address and store it for further use in RAM #1 38 as shown in Figure 4.

10 • If a pointer adjustment is detected than add or subtract 8 to the number representing the bits in FIFO 12 related to the pointer adjustment, stored in RAM #2 40 for the specific channel being processed, otherwise, if no pointer adjustment is detected, add zero. This is shown in Figure 5.

15 • Subtract the current number of bits in FIFO related to the pointer adjustment, content of RAM #2 40, from current FIFO write and read address difference, content of RAM #1 38, creating an effective FIFO address difference, overwriting the current FIFO write and read address difference address in RAM #1 38. This is
20 shown in Figure 6.

- Add the effective FIFO address difference, from RAM #1 38, to the accumulated effective FIFO address difference, stored in RAM #2 40, and after the addition store new value in the RAM #2 40 for further use as shown in Figure 7.

25 • Add the current effective FIFO address difference to the scaled accumulated value of the effective FIFO address difference. The scaling is performed by taking only the most significant bits of the word representing the accumulated

5 effective FIFO address difference. The multiplex circuit, MUX #3
46, is used for the scaling operation. This operation is
equivalent to implementation of a loop filter with an ideal
integrator and compensation. The ratio of accumulated effective
FIFO address difference value versus the current effective FIFO
10 address difference sets the position of the zero in the transfer
function of the loop filter. Without any disadvantage to the
performance, the ratio can be chosen to be power of 2 so a simple
multiplex circuit can accomplish scaling. The result can be
stored in the spare address RAM #2 40 location. This is shown in
15 Figure 8.

• Scale the product of the previous operation. This can be
again done by using a portion of bits that form the value after
the filtering action. This operation sets the Open Loop Gain of
the PLL, and therefore the bandwidth of the PLL. The used
20 topology allows for some useful features. Because the actual
FIFO address difference is calculated in the first step, the Open
Loop Gain can be modified to prevent the FIFO spill. An adaptive
gain with hysteresis is possible because the current gain value
can be stored in small latch in the control circuit 48 and we can
25 set the break-points for gain changes depending on address
difference as well as current value of the gain. If the FIFO
address difference is approaching its limits than the gain can be
increased to prevent a slip. This is shown in Figure 9. The
result of this process is a PLL phase increment value.

5 • The next step implements the semi-open loop handling of the bits in FIFO **12** due to pointer adjustments. A small number is added/subtracted to the PLL phase increment value. This small number is stored in RAM #1 **38** when the RAM is initialized, and it provides an extra phase modulation of the output clock designed

10 so it will leak the bits stored in FIFO **12** related to the pointer adjustment. The value of the external phase modulation can be changed. The best performance is achieved if the value is gradually increased so as to prevent discontinuities in the phase function as well as the first derivative of the phase function

15 (frequency). In this way, the bandwidth of the jitter is reduced. This is shown in Figure **10**.

• The last step is to update the number of bits in FIFO **12** due to the pointer adjustment stored in RAM #2 **40** by adding/subtracting the amount of external phase increment used to

20 extra modulate the endless phase modulator block **24**. Alternatively, this external phase increment value is converted to UI (user interval). This is shown in Figure **11**.

Figures **4** to **11** show the performance of the steps of a preferred embodiment. The structure is flexible so with a

25 minimum change in control signals, a different algorithm than that shown can be used.

After the total phase increment is obtained for an individual channel its value is written to the endless phase modulator block **24**. The endless phase modulator common control

5 block 71 is shown in Figure 12. The RAM #3 50 in Figure 12 is
used to store the total phase increment. Inside the endless
phase modulator block 24 the total phase increment value is added
to the current phase contained in RAM #4 52. This operation
could have been performed in the arithmetic unit block 18, but it
10 is better to perform it in the endless phase modulator block 24,
because, by doing so, intrinsic jitter is improved. The current
phase value is used to control the output of the endless phase
modulator 24. The current phase value is used as an address for
a Look-Up Table (LUT) in RAM #5 54. The LUT output yields the
15 setting for the multiplexer circuit select signal. The LUT is
built using a dual port RAM. The LUT values contain the select
values for the multiplexer circuits 56 in the endless modulator
delay line circuit 51 (Figure 13), and these values change with
process variations, because the unit delay will change perhaps
20 even +/-50 % from a nominal value depending on process variation.
If the current phase word is 8 bits wide one would need a LUT
that is 8 times the number of all possible numbers of delay taps
that are equal to one clock cycle. This would require a large
LUT so it is better to use a RAM as an LUT.

25 The content of the LUT is generated and changed on the
fly, depending also on the temperature variations. This is
achieved using a calibration circuit 55 shown in Figure 14. The
calibration circuit 55 determines the number of taps that form a
delay closest to one period of the crystal oscillator 26. Once
30 this number is known, it is easy to determine the content of the

5 LUT arithmetically. For example, if there are 73 taps that
constitute the delay of one clock cycle and 8 bits represent a
phase word, then the content of the LUT at address 73 should be
an increment of $73/256$. For the purpose of adding the number
representing the increment $73/256$, the adder circuit **53** in the
10 endless phase modulator common control block **24** is used. For
example, if a circuit is desynchronizing 12 DS3 channels dropped
from the OC-3 SONET signal, the adder **53** can perform adding of
the phase for each of the channels in round robin fashion and, at
the end of the cycle, it can update one address of the LUT.
15 After 256 round robin cycles, the whole LUT will be updated. The
time interval for these 256 cycles is a fraction of a second, a
much smaller time constant than the time constant of the
temperature variation process. The LUT contains in its address
space the incremental setting of the endless phase modulator
20 circuit select signal per one bit increment of the phase value.

The endless phase modulator block **24** consists of a
number of delay lines built using buffers. The total length of
delay should be longer than one clock cycle even for the worst
25 case of the process variation and the temperature change. A
calibration circuit **55** should be used to determine the number of
delay taps that is closest to the clock interval. The
calibration circuit **55** is operating all the time to adjust to
changes in temperature. Assume that there is N delay taps that
30 give a delay of almost one clock period. Each tap is worth $360/N$
degrees. The phase value is converted to tap number. The phase

5 value is compared to the multiples of tap value to determine
which tap should be activated. The proposed structure allows the
use of only one calibration circuit **55** and one delay line. The
buffers each have the same delay and track each other as far as
process variation as well as temperature. Therefore, one can use
10 one delay line to generate all clocks. Only the multiplex
circuits are required to be individual circuits for each channel.

When desynchronizing low data rate signals, like DS1
and E1, it may be more efficient to use the dual modulus divider
15 for large phase steps and the short delay line to generate small
phase steps. For the high data rates, like DS3 and E3, the whole
delay line can be made out of buffers. Because the delay depends
on temperature, the delay line needs to be constantly calibrated.
Note that in this implementation of the endless phase modulator
20 block **24** the loop bandwidth does not depend on the number of
delay elements in one clock period. Only the intrinsic jitter
depends on the number of delay elements in one clock cycle. The
intrinsic jitter, when measured with a 10 Hz high pass filter,
also depends on the frequency offset between the original clock
25 and the synthesized clock using the endless phase modulator block
24.

In the calibration circuit **55**, shown in Figure **14**,
multiplexer **81** receives as inputs the outputs from each of the
delay taps and selects the first tap to be outputted to flip flop
30 **80**. Multiplexer **83** receives the outputs from each tap and

5 outputs to flip flop 85 in accordance with the counter value from
counter 82 on line 84 coupled to the select input of multiplexer
83. With the rising edge of the clock from the first tap
selected from multiplexer 81, the output of flip flop 80 goes
high. The rising edge of the delayed clock selected by
10 multiplexer 83 by the count of counter 82 on line 84 causes flip
flop 85 to issue a positive output which resets flip flop 80 and
causes its output to go low. Once a clock from the Nth tap
number is selected, whose delay exceeds the time at which the
first tap clock falls back to a low, α remains high until the
15 clock signal from the Nth tap falls to zero. However, β remains
high disabling AND circuit 91 until the tap number is increased
so that a delay is larger than a whole cycle. At such a point, β
falls to zero and the transition is detected by flip flop 89, and
AND circuit 91 which operate as a negative edge detector. The
20 edge detection signal γ clocks in the count in counter 82 to
register 92 and latches the latter.

Referring to Figure 16, the waveform for the clock from
the first tap number is shown at the top. The waveform for α
shows it rising to "one" at the rising edge of the delayed clock
25 of the first tap number and staying there until the rising edge
of the Nth tap number delayed clock. In this case β remains low.
The bottom three waveforms show α for a high tap number in which
the rising edge of the Nth tap number delayed clock occurs after
the fall of the first tap number delayed clock. In this case α

5 remains at one and only resets when the rising edge of the Nth tap number clock. In this case β stays at one. Once the tap number is increased so that a delay is larger than a whole cycle, signal β is changed from high to low and the transition signal γ causes the current tap number to be latched in the register 92.

10 The latched tap number in register 92 is used to create a corrected frequency of operation. For example, assume that the tap number 16 produces a delay that is longer than one cycle. The latched signal that is given to register 92 by counter 82 is one less than 16, or 15 as the counter is started from the second tap. The number 15 is accumulated 255 times in the adder 53 (see Fig. 12), the register 57 and RAM #4 52. The products of this accumulation are written to RAM #5 54 by changing the write addresses of RAM #5 54 from 0 to 255. To update the content of RAM #5 54, the output of register 57 is fed to the data input of RAM #5 54. The numbers written into the data input of RAM #5 54 starting from address 0 to address 255 are as follows:

0, 15, 30, 60,, 3795, 3810, 3825

If the eight least significant bits from register 57 are ignored so that only the "M" most significant bits are transferred to the Data input of RAM #5 54, the output Select Tap No. will take on the following values:

0, 0, 0, 0,, 1, 1, 1,, 2, 2, 2,, 15, 15, 15, 15

5 When the foregoing values of Select Tap No. versus corresponding
data value are plotted one gets the staircase curve shown in
Figure 17. For comparison, the plot for a tap number of 20 is
shown in Figure 18. Thus, as all of the addresses in RAM #5 54
are cycled through from 0 to 255, one gets progressively
10 increasing Select Tap Nos. from the output of RAM #5 54 which are
applied to the gates of the multiplexers 56 to select the
corresponding delayed clock line. With crystal oscillator 26
running at approximately 44.736 MHz for DS3 desynchronization,
and using a frequency of 72 kHz for the updating the selection of
15 delay taps, there will be only 14 to 15 updates of the phase per
cycle of a frequency offset of 5 kHz. Physically, at each
successively larger tap selection the delayed clock originating
from the crystal oscillator 26 will be delayed 1/14 per one clock
cycle. With the addresses in RAM #5 54 progressively increasing
20 from 0 to 255 in about 14 microseconds, one less cycle of the
output clock is produced in this time. The foregoing adjustments
are repeated for a new value latched by the calibration circuit.
If the addresses of RAM #5 54 were progressively decreasing over
this time, this would cause the output clock to advance one whole
25 cycle so as to produce a higher output frequency.

The reason for adding the total phase increment in
endless phase modulator block 24, and not in the arithmetic unit
block 18, is due to the fact that this block has a smaller number
30 of steps and can perform the function of adding the total phase
increment to phase at a much higher rate than 72 kHz, as it might

5 be done in the arithmetic unit block **18**. A shorter interval
between the updates of delay taps selection will result in
smaller jitter. For example, in the case of DS3
desynchronization, if one needs to synthesize frequency offset of
5 kHz from the 44.736 MHz frequency of the crystal oscillator **26**,
10 using a 72 kHz update, one can have only 14 to 15 updates of the
phase per one output clock cycle. This will result in phase
steps of approximately $1/14$ UI. If the phase is updated at a
higher rate, for example at 500 kHz, the number of delay tap
updates per one clock cycle is at least 100 and the phase step
15 will be $1/100$ UI. Because the selected delay taps will be closer
to each other, glitching will be eliminated. When
desynchronizing low data rate signals, like DS1 and E1, it may be
more efficient to use the dual modulus divider for large phase
steps and the short delay line to generate small phase steps.
20 For the high data rates, like DS3 and E3, the whole delay line
can be made out of buffers. Because the delay depends on
temperature, the delay line needs to be constantly calibrated.
Note that in the present implementation of the endless phase
modulator block **24**, the loop bandwidth does not depend on the
25 number of delay elements in one clock period. Only the intrinsic
jitter depends on the number of delay elements in one clock
cycle. The intrinsic jitter, when measured with a 10 Hz high
pass filter, also depends on the frequency offset between the
original clock and the synthesized clock using the endless phase
30 modulator block **24**.

5 The alternative embodiment of Figure 15 shows a pointer
adjustment signal that can be derived in a de-mapper so a pointer
adjustment block may not be required. It also shows the system
consisting of a numerically controlled oscillator (NCO) block 70,
an I and Q DAC 72 and a single side-band modulator (SSB) 74 that
10 can be used to effectively create the high frequency digitally
controlled oscillator. The NCO block 70 looks exactly like the
endless phase modulator block 24 except the RAM based LUT can be
replaced by a sinusoidal ROM (not shown). The calculated phase
is used as an address to the sinusoidal ROM to convert phase to
15 amplitude. Only one quarter of the sinusoid needs to be stored,
by inverting amplitude or address, the other three quarters of
the sinusoid can be created. In the NCO block 70 the sine and
cosine amplitude digital values are then converted to analog
values with the use of a couple of one bit fractional digital to
20 Analog Converter (DAC) circuits (not shown). These I and Q
signals are then up-converted to the proper frequency with use of
the SSB modulator 74.

 Accordingly, while this invention has been described
25 with reference to illustrative embodiments, this description is
not intended to be construed in a limiting sense. Various
modifications of the illustrative embodiments, as well as other
embodiments of the invention, will be apparent to persons skilled
in the art upon reference to this description. It is therefore
30 contemplated that the appended claims will cover any such

5 modifications or embodiments as fall within the true scope of the invention.